# 2. Burglary

**Program Name: Burglary.java** **Input File: burglary.dat**

You are organizing a heist of a large bank, and have obtained a blueprint of the bank's vault. The vault consists of a series of N rooms connected in a row. You start in the left-most room. Each room contains a certain amount of money and a vault door connecting it to the next room. The blueprints detail the amount of time it would take you to crack each vault door. You realize that if you take just the money from the first room and leave immediately, the police will not be able to catch you, and you will get to keep all of the money you took, But, if you try to open any vault doors, there is a 5% higher chance per door that the police will show up and you would have to dump the money and run, leaving you with nothing. For example, if you take the time to open 3 vault doors, there is a 15% chance the police will show up. Naturally, being a risky person, you do not want to take the safe route of only taking the money from the first room. You want to maximize the expected value of your heist. The expected value of a given heist plan is the probability you keep the money multiplied by the amount of money you stole. If the probability of the police showing up is p, then the probability of the police not showing up is (1-p), and if you would steal M dollars if you got away, then the expected value of that heist plan is (1-p)*M.

How many doors should you open to maximize your expected value? And what is the maximum expected value of money you can steal?

### Input
The first line of input contains T, the number of test cases that follow.

The first line of each test case is the number of rooms N. The next line contains N space separated money values, the amount of money in each room.

### Output
For each test case, output the number of doors to open to maximize your expected value, and the expected value of opening that many doors, separated by a space. The expected value should be rounded to the nearest cent.

### Constraints
```
1 <= T <= 8
1 <= N <= 40
```

### Example Input File
```
3
3
5.00 5.00 5.00
2
100.00 1.00
5
44.90 4.95 60.02 26.65 90.87
```

### Example Output to Screen
```
2 13.50
0 100.00
4 181.91
```

### Explanation of Output
In the first test case, if you do not open any doors, and just take the money in the first room, the expected value is 1.00 * 5.00 = 5.00. If you open the first door, and then run, the expected value is 0.95 * 10.00 = 9.50. If you open the first door, and then the second door, the expected value is 0.90 * 15.00 = 13.50. The maximum of these is 2 doors with 13.50.

In the second test case, if you take the money and run, the expected value is 1.00 * 100.00 = 100.00. If you open the first door and then run, the expected value is 0.95 * 101.00 = 95.95. The maximum of these is 0 doors with 100.00.